The Web Application Hacker's Handbook

Second Edition

Finding and Exploiting Security Flaws

Dafydd Stuttard
Marcus Pinto

WILEY
Wiley Publishing, Inc.

The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Second Edition

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
`www.wiley.com`

Copyright © 2011 by Dafydd Stuttard and Marcus Pinto
Published by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-118-02647-2
ISBN: 978-1-118-17522-4 (ebk)
ISBN: 978-1-118-17524-8 (ebk)
ISBN: 978-1-118-17523-1 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at `http://www.wiley.com/go/permissions`.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or website may provide or recommendations it may make. Further, readers should be aware that Internet websites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Not all content that is available in standard print versions of this book may appear or be packaged in all book formats. If you have purchased a version of this book that did not include media that is referenced by or accompanies a standard print version, you may request this media by visiting `http://booksupport.wiley.com`. For more information about Wiley products, visit us at `www.wiley.com`.

Library of Congress Control Number: 2011934639

Trademarks: Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

About the Authors

Dafydd Stuttard is an independent security consultant, author, and software developer. With more than 10 years of experience in security consulting, he specializes in the penetration testing of web applications and compiled software. Dafydd has worked with numerous banks, retailers, and other enterprises to help secure their web applications. He also has provided security consulting to several software manufacturers and governments to help secure their compiled software. Dafydd is an accomplished programmer in several languages. His interests include developing tools to facilitate all kinds of software security testing. Under the alias "PortSwigger," Dafydd created the popular Burp Suite of web application hacking tools; he continues to work actively on Burp's development. Dafydd is also cofounder of MDSec, a company providing training and consultancy on Internet security attack and defense. Dafydd has developed and presented training courses at various security conferences around the world, and he regularly delivers training to companies and governments. He holds master's and doctorate degrees in philosophy from the University of Oxford.

Marcus Pinto is cofounder of MDSec, developing and delivering training courses in web application security. He also performs ongoing security consultancy for financial, government, telecom, and retail verticals. His 11 years of experience in the industry have been dominated by the technical aspects of application security, from the dual perspectives of a consulting and end-user implementation role. Marcus has a background in attack-based security assessment and penetration testing. He has worked extensively with large-scale web application deployments in the financial services industry. Marcus has been developing and presenting database and web application training courses since 2005 at Black Hat and other worldwide security conferences, and for private-sector and government clients. He holds a master's degree in physics from the University of Cambridge.

About the Technical Editor

Dr. Josh Pauli received his Ph.D. in Software Engineering from North Dakota State University (NDSU) with an emphasis in secure requirements engineering and now serves as an Associate Professor of Information Security at Dakota State University (DSU). Dr. Pauli has published nearly 20 international journal and conference papers related to software security and his work includes invited presentations from the Department of Homeland Security and Black Hat Briefings. He teaches both undergraduate and graduate courses in system software security and web software security at DSU. Dr. Pauli also conducts web application penetration tests as a Senior Penetration Tester for an Information Security consulting firm where his duties include developing hands-on technical workshops in the area of web software security for IT professionals in the financial sector.

MDSec: The Authors' Company

Dafydd and Marcus are cofounders of MDSec, a company that provides training in attack and defense-based security, along with other consultancy services. If while reading this book you would like to put the concepts into practice, and gain hands-on experience in the areas covered, you are encouraged to visit our website, `http://mdsec.net`. This will give you access to hundreds of interactive vulnerability labs and other resources that are referenced throughout the book.

Credits

Executive Editor
Carol Long

Senior Project Editor
Adaobi Obi Tulton

Technical Editor
Josh Pauli

Production Editor
Kathleen Wisor

Copy Editor
Gayle Johnson

Editorial Manager
Mary Beth Wakefield

Freelancer Editorial Manager
Rosemarie Graham

Associate Director of Marketing
David Mayhew

Marketing Manager
Ashley Zurcher

Business Manager
Amy Knies

Production Manager
Tim Tate

Vice President and Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Neil Edde

Associate Publisher
Jim Minatel

Project Coordinator, Cover
Katie Crocker

Proofreaders
Sarah Kaikini, Word One
Sheilah Ledwidge, Word One

Indexer
Robert Swanson

Cover Designer
Ryan Sneed

Cover Image
Wiley InHouse Design

Vertical Websites Project Manager
Laura Moss-Hollister

Vertical Websites Assistant Project Manager
Jenny Swisher

Vertical Websites Associate Producers
Josh Frank
Shawn Patrick
Doug Kuhn
Marilyn Hummel

Acknowledgments

We are indebted to the directors and others at Next Generation Security Software, who provided the right environment for us to realize the first edition of this book. Since then, our input has come from an increasingly wider community of researchers and professionals who have shared their ideas and contributed to the collective understanding of web application security issues that exists today. Because this is a practical handbook rather than a work of scholarship, we have deliberately avoided filling it with a thousand citations of influential articles, books, and blog postings that spawned the ideas involved. We hope that people whose work we discuss anonymously are content with the general credit given here.

We are grateful to the people at Wiley — in particular, to Carol Long for enthusiastically supporting our project from the outset, to Adaobi Obi Tulton for helping polish our manuscript and coaching us in the quirks of "American English," to Gayle Johnson for her very helpful and attentive copy editing, and to Katie Wisor's team for delivering a first-rate production.

A large measure of thanks is due to our respective partners, Becky and Amanda, for tolerating the significant distraction and time involved in producing a book of this size.

Both authors are indebted to the people who led us into our unusual line of work. Dafydd would like to thank Martin Law. Martin is a great guy who first taught me how to hack and encouraged me to spend my time developing techniques and tools for attacking applications. Marcus would like to thank his parents for everything they have done and continue to do, including getting me into computers. I've been getting into computers ever since.

Contents at a Glance

Introduction xxiii
Chapter 1 Web Application (In)security 1
Chapter 2 Core Defense Mechanisms 17
Chapter 3 Web Application Technologies 39
Chapter 4 Mapping the Application 73
Chapter 5 Bypassing Client-Side Controls 117
Chapter 6 Attacking Authentication 159
Chapter 7 Attacking Session Management 205
Chapter 8 Attacking Access Controls 257
Chapter 9 Attacking Data Stores 287
Chapter 10 Attacking Back-End Components 357
Chapter 11 Attacking Application Logic 405
Chapter 12 Attacking Users: Cross-Site Scripting 431
Chapter 13 Attacking Users: Other Techniques 501
Chapter 14 Automating Customized Attacks 571
Chapter 15 Exploiting Information Disclosure 615
Chapter 16 Attacking Native Compiled Applications 633
Chapter 17 Attacking Application Architecture 647
Chapter 18 Attacking the Application Server 669
Chapter 19 Finding Vulnerabilities in Source Code 701
Chapter 20 A Web Application Hacker's Toolkit 747
Chapter 21 A Web Application Hacker's Methodology 791
Index 853

Contents

Introduction **xxiii**

Chapter 1 **Web Application (In)security** **1**
The Evolution of Web Applications 2
  Common Web Application Functions 4
  Benefits of Web Applications 5
Web Application Security 6
  "This Site Is Secure" 7
  The Core Security Problem: Users Can Submit Arbitrary Input 9
  Key Problem Factors 10
  The New Security Perimeter 12
  The Future of Web Application Security 14
Summary 15

Chapter 2 **Core Defense Mechanisms** **17**
Handling User Access 18
  Authentication 18
  Session Management 19
  Access Control 20
Handling User Input 21
  Varieties of Input 21
  Approaches to Input Handling 23
  Boundary Validation 25
  Multistep Validation and Canonicalization 28
Handling Attackers 30
  Handling Errors 30
  Maintaining Audit Logs 31
  Alerting Administrators 33
  Reacting to Attacks 34

Managing the Application 35
Summary 36
Questions 36

Chapter 3 Web Application Technologies 39
The HTTP Protocol 39
  HTTP Requests 40
  HTTP Responses 41
  HTTP Methods 42
  URLs 44
  REST 44
  HTTP Headers 45
  Cookies 47
  Status Codes 48
  HTTPS 49
  HTTP Proxies 49
  HTTP Authentication 50
Web Functionality 51
  Server-Side Functionality 51
  Client-Side Functionality 57
  State and Sessions 66
Encoding Schemes 66
  URL Encoding 67
  Unicode Encoding 67
  HTML Encoding 68
  Base64 Encoding 69
  Hex Encoding 69
  Remoting and Serialization Frameworks 70
Next Steps 70
Questions 71

Chapter 4 Mapping the Application 73
Enumerating Content and Functionality 74
  Web Spidering 74
  User-Directed Spidering 77
  Discovering Hidden Content 80
  Application Pages Versus Functional Paths 93
  Discovering Hidden Parameters 96
Analyzing the Application 97
  Identifying Entry Points for User Input 98
  Identifying Server-Side Technologies 101
  Identifying Server-Side Functionality 107
  Mapping the Attack Surface 111
Summary 114
Questions 114

Chapter 5 Bypassing Client-Side Controls 117

Transmitting Data Via the Client 118
  Hidden Form Fields 118
  HTTP Cookies 121
  URL Parameters 121
  The Referer Header 122
  Opaque Data 123
  The ASP.NET ViewState 124
Capturing User Data: HTML Forms 127
  Length Limits 128
  Script-Based Validation 129
  Disabled Elements 131
Capturing User Data: Browser Extensions 133
  Common Browser Extension Technologies 134
  Approaches to Browser Extensions 135
  Intercepting Traffic from Browser Extensions 135
  Decompiling Browser Extensions 139
  Attaching a Debugger 151
  Native Client Components 153
Handling Client-Side Data Securely 154
  Transmitting Data Via the Client 154
  Validating Client-Generated Data 155
  Logging and Alerting 156
Summary 156
Questions 157

Chapter 6 Attacking Authentication 159

Authentication Technologies 160
Design Flaws in Authentication Mechanisms 161
  Bad Passwords 161
  Brute-Forcible Login 162
  Verbose Failure Messages 166
  Vulnerable Transmission of Credentials 169
  Password Change Functionality 171
  Forgotten Password Functionality 173
  “Remember Me” Functionality 176
  User Impersonation Functionality 178
  Incomplete Validation of Credentials 180
  Nonunique Usernames 181
  Predictable Usernames 182
  Predictable Initial Passwords 183
  Insecure Distribution of Credentials 184
Implementation Flaws in Authentication 185
  Fail-Open Login Mechanisms 185
  Defects in Multistage Login Mechanisms 186
  Insecure Storage of Credentials 190

Securing Authentication 191
  Use Strong Credentials 192
  Handle Credentials Secretively 192
  Validate Credentials Properly 193
  Prevent Information Leakage 195
  Prevent Brute-Force Attacks 196
  Prevent Misuse of the Password Change Function 199
  Prevent Misuse of the Account Recovery Function 199
  Log, Monitor, and Notify 201
Summary 201
Questions 202

Chapter 7 Attacking Session Management 205
The Need for State 206
  Alternatives to Sessions 208
Weaknesses in Token Generation 210
  Meaningful Tokens 210
  Predictable Tokens 213
  Encrypted Tokens 223
Weaknesses in Session Token Handling 233
  Disclosure of Tokens on the Network 234
  Disclosure of Tokens in Logs 237
  Vulnerable Mapping of Tokens to Sessions 240
  Vulnerable Session Termination 241
  Client Exposure to Token Hijacking 243
  Liberal Cookie Scope 244
Securing Session Management 248
  Generate Strong Tokens 248
  Protect Tokens Throughout Their Life Cycle 250
  Log, Monitor, and Alert 253
Summary 254
Questions 255

Chapter 8 Attacking Access Controls 257
Common Vulnerabilities 258
  Completely Unprotected Functionality 259
  Identifier-Based Functions 261
  Multistage Functions 262
  Static Files 263
  Platform Misconfiguration 264
  Insecure Access Control Methods 265
Attacking Access Controls 266
  Testing with Different User Accounts 267
  Testing Multistage Processes 271
  Testing with Limited Access 273
  Testing Direct Access to Methods 276
  Testing Controls Over Static Resources 277

Testing Restrictions on HTTP Methods 278
Securing Access Controls 278
A Multilayered Privilege Model 280
Summary 284
Questions 284

Chapter 9 Attacking Data Stores 287
Injecting into Interpreted Contexts 288
Bypassing a Login 288
Injecting into SQL 291
Exploiting a Basic Vulnerability 292
Injecting into Different Statement Types 294
Finding SQL Injection Bugs 298
Fingerprinting the Database 303
The UNION Operator 304
Extracting Useful Data 308
Extracting Data with UNION 308
Bypassing Filters 311
Second-Order SQL Injection 313
Advanced Exploitation 314
Beyond SQL Injection: Escalating the Database Attack 325
Using SQL Exploitation Tools 328
SQL Syntax and Error Reference 332
Preventing SQL Injection 338
Injecting into NoSQL 342
Injecting into MongoDB 343
Injecting into XPath 344
Subverting Application Logic 345
Informed XPath Injection 346
Blind XPath Injection 347
Finding XPath Injection Flaws 348
Preventing XPath Injection 349
Injecting into LDAP 349
Exploiting LDAP Injection 351
Finding LDAP Injection Flaws 353
Preventing LDAP Injection 354
Summary 354
Questions 354

Chapter 10 Attacking Back-End Components 357
Injecting OS Commands 358
Example 1: Injecting Via Perl 358
Example 2: Injecting Via ASP 360
Injecting Through Dynamic Execution 362
Finding OS Command Injection Flaws 363
Finding Dynamic Execution Vulnerabilities 366

Preventing OS Command Injection 367
Preventing Script Injection Vulnerabilities 368
Manipulating File Paths 368
Path Traversal Vulnerabilities 368
File Inclusion Vulnerabilities 381
Injecting into XML Interpreters 383
Injecting XML External Entities 384
Injecting into SOAP Services 386
Finding and Exploiting SOAP Injection 389
Preventing SOAP Injection 390
Injecting into Back-end HTTP Requests 390
Server-side HTTP Redirection 390
HTTP Parameter Injection 393
Injecting into Mail Services 397
E-mail Header Manipulation 398
SMTP Command Injection 399
Finding SMTP Injection Flaws 400
Preventing SMTP Injection 402
Summary 402
Questions 403

Chapter 11 Attacking Application Logic 405
The Nature of Logic Flaws 406
Real-World Logic Flaws 406
Example 1: Asking the Oracle 407
Example 2: Fooling a Password Change Function 409
Example 3: Proceeding to Checkout 410
Example 4: Rolling Your Own Insurance 412
Example 5: Breaking the Bank 414
Example 6: Beating a Business Limit 416
Example 7: Cheating on Bulk Discounts 418
Example 8: Escaping from Escaping 419
Example 9: Invalidating Input Validation 420
Example 10: Abusing a Search Function 422
Example 11: Snarfing Debug Messages 424
Example 12: Racing Against the Login 426
Avoiding Logic Flaws 428
Summary 429
Questions 430

Chapter 12 Attacking Users: Cross-Site Scripting 431
Varieties of XSS 433
Reflected XSS Vulnerabilities 434
Stored XSS Vulnerabilities 438
DOM-Based XSS Vulnerabilities 440
XSS Attacks in Action 442
Real-World XSS Attacks 442

    Payloads for XSS Attacks 443
    Delivery Mechanisms for XSS Attacks 447
Finding and Exploiting XSS Vulnerabilities 451
    Finding and Exploiting Reflected XSS Vulnerabilities 452
    Finding and Exploiting Stored XSS Vulnerabilities 481
    Finding and Exploiting DOM-Based XSS Vulnerabilities 487
Preventing XSS Attacks 492
    Preventing Reflected and Stored XSS 492
    Preventing DOM-Based XSS 496
Summary 498
Questions 498

Chapter 13 Attacking Users: Other Techniques 501
Inducing User Actions 501
    Request Forgery 502
    UI Redress 511
Capturing Data Cross-Domain 515
    Capturing Data by Injecting HTML 516
    Capturing Data by Injecting CSS 517
    JavaScript Hijacking 519
The Same-Origin Policy Revisited 524
    The Same-Origin Policy and Browser Extensions 525
    The Same-Origin Policy and HTML5 528
    Crossing Domains with Proxy Service Applications 529
Other Client-Side Injection Attacks 531
    HTTP Header Injection 531
    Cookie Injection 536
    Open Redirection Vulnerabilities 540
    Client-Side SQL Injection 547
    Client-Side HTTP Parameter Pollution 548
Local Privacy Attacks 550
    Persistent Cookies 550
    Cached Web Content 551
    Browsing History 552
    Autocomplete 552
    Flash Local Shared Objects 553
    Silverlight Isolated Storage 553
    Internet Explorer userData 554
    HTML5 Local Storage Mechanisms 554
    Preventing Local Privacy Attacks 554
Attacking ActiveX Controls 555
    Finding ActiveX Vulnerabilities 556
    Preventing ActiveX Vulnerabilities 558
Attacking the Browser 559
    Logging Keystrokes 560
    Stealing Browser History and Search Queries 560

Enumerating Currently Used Applications 560
Port Scanning 561
Attacking Other Network Hosts 561
Exploiting Non-HTTP Services 562
Exploiting Browser Bugs 563
DNS Rebinding 563
Browser Exploitation Frameworks 564
Man-in-the-Middle Attacks 566
Summary 568
Questions 568

Chapter 14 Automating Customized Attacks 571
Uses for Customized Automation 572
Enumerating Valid Identifiers 573
The Basic Approach 574
Detecting Hits 574
Scripting the Attack 576
JAttack 577
Harvesting Useful Data 583
Fuzzing for Common Vulnerabilities 586
Putting It All Together: Burp Intruder 590
Barriers to Automation 602
Session-Handling Mechanisms 602
CAPTCHA Controls 610
Summary 613
Questions 613

Chapter 15 Exploiting Information Disclosure 615
Exploiting Error Messages 615
Script Error Messages 616
Stack Traces 617
Informative Debug Messages 618
Server and Database Messages 619
Using Public Information 623
Engineering Informative Error Messages 624
Gathering Published Information 625
Using Inference 626
Preventing Information Leakage 627
Use Generic Error Messages 628
Protect Sensitive Information 628
Minimize Client-Side Information Leakage 629
Summary 629
Questions 630

Chapter 16 Attacking Native Compiled Applications 633
Buffer Overflow Vulnerabilities 634
Stack Overflows 634
Heap Overflows 635

- "Off-by-One" Vulnerabilities 636
- Detecting Buffer Overflow Vulnerabilities 639

Integer Vulnerabilities 640
- Integer Overflows 640
- Signedness Errors 641
- Detecting Integer Vulnerabilities 642

Format String Vulnerabilities 643
- Detecting Format String Vulnerabilities 644

Summary 645

Questions 645

Chapter 17 Attacking Application Architecture 647

Tiered Architectures 647
- Attacking Tiered Architectures 648
- Securing Tiered Architectures 654

Shared Hosting and Application Service Providers 656
- Virtual Hosting 657
- Shared Application Services 657
- Attacking Shared Environments 658
- Securing Shared Environments 665

Summary 667

Questions 667

Chapter 18 Attacking the Application Server 669

Vulnerable Server Configuration 670
- Default Credentials 670
- Default Content 671
- Directory Listings 677
- WebDAV Methods 679
- The Application Server as a Proxy 682
- Misconfigured Virtual Hosting 683
- Securing Web Server Configuration 684

Vulnerable Server Software 684
- Application Framework Flaws 685
- Memory Management Vulnerabilities 687
- Encoding and Canonicalization 689
- Finding Web Server Flaws 694
- Securing Web Server Software 695

Web Application Firewalls 697

Summary 699

Questions 699

Chapter 19 Finding Vulnerabilities in Source Code 701

Approaches to Code Review 702
- Black-Box Versus White-Box Testing 702
- Code Review Methodology 703

Signatures of Common Vulnerabilities 704
- Cross-Site Scripting 704

SQL Injection 705
Path Traversal 706
Arbitrary Redirection 707
OS Command Injection 708
Backdoor Passwords 708
Native Software Bugs 709
Source Code Comments 710
The Java Platform 711
Identifying User-Supplied Data 711
Session Interaction 712
Potentially Dangerous APIs 713
Configuring the Java Environment 716
ASP.NET 718
Identifying User-Supplied Data 718
Session Interaction 719
Potentially Dangerous APIs 720
Configuring the ASP.NET Environment 723
PHP 724
Identifying User-Supplied Data 724
Session Interaction 727
Potentially Dangerous APIs 727
Configuring the PHP Environment 732
Perl 735
Identifying User-Supplied Data 735
Session Interaction 736
Potentially Dangerous APIs 736
Configuring the Perl Environment 739
JavaScript 740
Database Code Components 741
SQL Injection 741
Calls to Dangerous Functions 742
Tools for Code Browsing 743
Summary 744
Questions 744

Chapter 20 A Web Application Hacker's Toolkit 747
Web Browsers 748
Internet Explorer 748
Firefox 749
Chrome 750
Integrated Testing Suites 751
How the Tools Work 751
Testing Work Flow 769
Alternatives to the Intercepting Proxy 771
Standalone Vulnerability Scanners 773
Vulnerabilities Detected by Scanners 774
Inherent Limitations of Scanners 776

Technical Challenges Faced by Scanners 778
Current Products 781
Using a Vulnerability Scanner 783
Other Tools 785
Wikto/Nikto 785
Firebug 785
Hydra 785
Custom Scripts 786
Summary 789

Chapter 21 A Web Application Hacker's Methodology 791
General Guidelines 793
1 Map the Application's Content 795
1.1 Explore Visible Content 795
1.2 Consult Public Resources 796
1.3 Discover Hidden Content 796
1.4 Discover Default Content 797
1.5 Enumerate Identifier-Specified Functions 797
1.6 Test for Debug Parameters 798
2 Analyze the Application 798
2.1 Identify Functionality 798
2.2 Identify Data Entry Points 799
2.3 Identify the Technologies Used 799
2.4 Map the Attack Surface 800
3 Test Client-Side Controls 800
3.1 Test Transmission of Data Via the Client 801
3.2 Test Client-Side Controls Over User Input 801
3.3 Test Browser Extension Components 802
4 Test the Authentication Mechanism 805
4.1 Understand the Mechanism 805
4.2 Test Password Quality 806
4.3 Test for Username Enumeration 806
4.4 Test Resilience to Password Guessing 807
4.5 Test Any Account Recovery Function 807
4.6 Test Any Remember Me Function 808
4.7 Test Any Impersonation Function 808
4.8 Test Username Uniqueness 809
4.9 Test Predictability of Autogenerated Credentials 809
4.10 Check for Unsafe Transmission of Credentials 810
4.11 Check for Unsafe Distribution of Credentials 810
4.12 Test for Insecure Storage 811
4.13 Test for Logic Flaws 811
4.14 Exploit Any Vulnerabilities to Gain Unauthorized Access 813
5 Test the Session Management Mechanism 814
5.1 Understand the Mechanism 814
5.2 Test Tokens for Meaning 815
5.3 Test Tokens for Predictability 816

5.4 Check for Insecure Transmission of Tokens 817
5.5 Check for Disclosure of Tokens in Logs 817
5.6 Check Mapping of Tokens to Sessions 818
5.7 Test Session Termination 818
5.8 Check for Session Fixation 819
5.9 Check for CSRF 820
5.10 Check Cookie Scope 820
6 Test Access Controls 821
6.1 Understand the Access Control Requirements 821
6.2 Test with Multiple Accounts 822
6.3 Test with Limited Access 822
6.4 Test for Insecure Access Control Methods 823
7 Test for Input-Based Vulnerabilities 824
7.1 Fuzz All Request Parameters 824
7.2 Test for SQL Injection 827
7.3 Test for XSS and Other Response Injection 829
7.4 Test for OS Command Injection 832
7.5 Test for Path Traversal 833
7.6 Test for Script Injection 835
7.7 Test for File Inclusion 835
8 Test for Function-Specific Input Vulnerabilities 836
8.1 Test for SMTP Injection 836
8.2 Test for Native Software Vulnerabilities 837
8.3 Test for SOAP Injection 839
8.4 Test for LDAP Injection 839
8.5 Test for XPath Injection 840
8.6 Test for Back-End Request Injection 841
8.7 Test for XXE Injection 841
9 Test for Logic Flaws 842
9.1 Identify the Key Attack Surface 842
9.2 Test Multistage Processes 842
9.3 Test Handling of Incomplete Input 843
9.4 Test Trust Boundaries 844
9.5 Test Transaction Logic 844
10 Test for Shared Hosting Vulnerabilities 845
10.1 Test Segregation in Shared Infrastructures 845
10.2 Test Segregation Between ASP-Hosted Applications 845
11 Test for Application Server Vulnerabilities 846
11.1 Test for Default Credentials 846
11.2 Test for Default Content 847
11.3 Test for Dangerous HTTP Methods 847
11.4 Test for Proxy Functionality 847
11.5 Test for Virtual Hosting Misconfiguration 847
11.6 Test for Web Server Software Bugs 848
11.7 Test for Web Application Firewalling 848

12 Miscellaneous Checks 849
    12.1 Check for DOM-Based Attacks 849
    12.2 Check for Local Privacy Vulnerabilities 850
    12.3 Check for Weak SSL Ciphers 851
    12.4 Check Same-Origin Policy Configuration 851
13 Follow Up Any Information Leakage 852

Index **853**